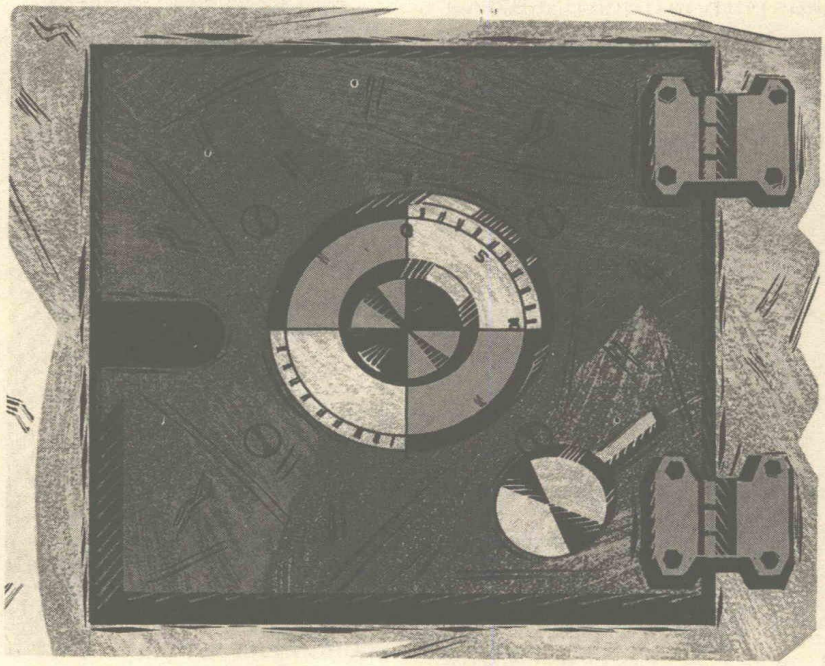
สมชัย แก้วศิริรัตน์

เทคนิคการตรวจสอบ ข้อผิดพลาด ในการเขียนโปรแกรม (ปาสคาล)

แนะนำวิธีการตรวจสอบหาตำแหน่งของจุดบกพร่องในโปรแกรมเพื่อทำการแก้ไข



ถ้าเราสังเกตดูในเมนูของเทอร์โบ-ปาสคาลจะพบว่ามีรายการให้ผู้ใช้สามารถที่จะตรวจสอบค่าของตัวแปรต่าง ๆ ในโปรแกรมว่าเปลี่ยนแปลงไปอย่างไรเมื่อโปรแกรมถูกทำงานทีละคำสั่ง ทั้งนี้อาศัยรายการ 2 รายการดังนี้คือ

1. RUN/TRACE INTO หรือ F7
2. BREAK WATCH/ADD WATCH หรือ Ctrl + F7

โดยที่รายการที่ 1 เป็นการกำหนดให้โปรแกรมทำงานทีละคำสั่ง ส่วนรายการที่ 2 เป็นการเพิ่มตัวแปรที่ต้องการทราบค่าเมื่อโปรแกรมเปลี่ยนไป ขั้นตอนที่ต้องกระทำคือ เลือกรายการที่ 2 แล้วระบุตัวแปรที่เราสนใจจนครบทุกตัว จากนั้นจึงเลือกรายการที่ 1 เพื่อดูค่าของตัวแปรที่เปลี่ยนไปหลังจากที่คำสั่งแต่ละคำสั่งถูกกระทำ ก่อนอื่นขอให้พิจารณาโปรแกรมตัวอย่างในรูปที่ 1 ซึ่งเป็นโปรแกรมตรวจสอบตำแหน่งที่ผิด

ตรวจสอบโปรแกรม

จากรูปที่ 1 โปรแกรมที่ยกตัวอย่างมานี้มีข้อผิดพลาดในการเขียนโปรแกรม (Logic error) ทั้งนี้เนื่องจากต้องการตรวจสอบค่าของตัวแปรต่าง ๆ ว่าผิดอย่างไร เพื่อที่จะได้ทราบตำแหน่งที่ผิด โดยเริ่มต้นดังนี้ เลือกรายการ RUN จากนั้นจึงเลือกรายการย่อย TRACE INTO หรือใช้วิธีการกดปุ่ม F7 แทนการเลือกรายการทั้งสองเพื่อกำหนดจุดเริ่มต้นให้โปรแกรมทำงาน ส่วนตัวแปรที่สนใจได้แก่ 1, c, top, min, i และ k หลังจากที่เลือกรายการทั้งสองแล้วจะปรากฏแถบสว่างที่บรรทัด begin ซึ่งเป็นคำสั่งแรกในโปรแกรม ให้เลื่อนเคอร์เซอร์ไปยังตัวแปรที่สนใจ ซึ่งในขั้นแรกให้เลื่อนเคอร์เซอร์ไปที่ตัวแปร 1 จากนั้นให้กด Ctrl กับ F7 หรือเลือกรายการ BREAK/WATCH แล้วจึงเลือกรายการย่อย ADD WATCH จะปรากฏตัวแปร 1 อยู่ในกรอบของ WATCH ให้กดปุ่ม <ENTER> เพื่อรับตัวแปรดังกล่าว จากนั้นให้เลื่อนเคอร์เซอร์ไปที่ตัวแปรอื่นที่สนใจแล้วกด Ctrl กับ F7 และตามด้วย <ENTER> จนกว่าจะครบทุกตัวแปร

```
Program test;
var c,top,min,k:word;
      temp,i : integer;
      l : array[1..10] of integer;
begin
      c := 10;
      for i := 1 to c do
         l[i] := random(100);
      for top := 1 to c-1 do
      begin
         min := top;
         for k := top + 1 to c do
            if l[k] < l[min] then
   l[min] := l[k];
  if min <> top then
  begin
    temp := l[top];
    l[top] := l[min];
    l[min] := temp;
  end;
end;
  end.
```

รูปที่ 1
ตัวอย่างโปรแกรมที่จะทำการตรวจสอบตำแหน่งที่ผิด

ขั้นตอนต่อไปเราต้องการที่จะทราบค่าของตัวแปรแต่ละตัวที่เปลี่ยนไป ให้กด F7 หรือเลือกรายการ RUN จากนั้นจึงเลือกรายการย่อย TRACE INTO เมื่อกดแต่ละครั้งแถบสว่างจะเลื่อนไปยังคำสั่งถัดไปพร้อมกับค่าของตัวแปรที่เลือกไว้จะเปลี่ยนไปเรื่อย ๆ โดยตอนเริ่มต้นนี้เมื่อกด F7 แต่ละครั้งจะเป็นการสร้างเลขสุ่มที่มีค่าไม่เกิน 100 จำนวน 10 ค่า และเมื่อกดไปเรื่อย ๆ จนกระทั่งแถบสว่างมาอยู่ที่บรรทัด if l[k] < l[min] then สังเกตว่าค่าของ k, min และ top เป็น 2, 1 และ 1 ตามลำดับ ให้กด F7 ไปเรื่อย ๆ จนกระทั่งหลุดลูปใน จะพบคำสั่ง if min <> top then แล้วย้อนกลับไปยังลูปนอก ให้กด F7 ไปเรื่อย ๆ จนกระทั่งมาที่บรรทัด if l[k] < l[min] then ที่จุดนี้ค่าในกรอบของ WATCH จะคล้ายดังนี้

```
  k : 3
min : 2
top : 2
  1 : (0, 23, 11, 51, 45,
       28, 10, 27, 39, 90)
```

ตำแหน่งที่ผิด

สังเกตว่าค่าของ min และ top เท่ากันทุกรอบ ซึ่งที่จริงแล้วค่าของ min ถูกกำหนดให้มีค่าเท่ากับ top ก่อนเริ่มเปรียบเทียบ และเมื่อมีการเปรียบเทียบและเป็นไปตามเงื่อนไขก็ไม่มีการเปลี่ยนแปลงค่าของ min เลย นั่นคือจะต้องมีข้อผิดพลาดเกิดขึ้นอย่างแน่นอน ไม่คำสั่งใดก็คำสั่งหนึ่ง

สำหรับข้อผิดพลาดในโปรแกรมอยู่ที่ตำแหน่ง if l[k] < l[min] then l[min] := l[k]; จะต้องเปลี่ยนเป็น if l[k] < l[min] then min := k; การที่จะแก้ไขโปรแกรมทำได้โดย เลือกรายการ RUN แล้วเลือกรายการย่อย Go to cursor หรือกดปุ่ม F4 แทนการเลือกรายการดังกล่าว ถ้าโปรแกรมมีการเปลี่ยนแปลงที่จอภาพจะปรากฏข้อความดังนี้ Source Modified, rebuild ? (Y/N) : ให้กด Y หลังจากนั้นโปรแกรมจะถูกคอมพายล์และรันใหม่ จากนั้นจึงแก้ไขโปรแกรมตรงตำแหน่งที่ผิดและทดลองตรวจสอบอีกจนกว่าจะถูกต้อง

การแก้ไขโปรแกรมที่มีขนาดใหญ่

ในบางครั้งจำเป็นต้องแก้ไขโปรแกรมที่มีขนาดใหญ่ โดยเฉพาะอย่างยิ่งกรณีที่โปรแกรมหนึ่ง ๆ ถูกแบ่งออกเป็นหลาย ๆ ส่วน การหาที่ผิดพลาดนั้นอาจไล่จนหลงทาง ซึ่งบางครั้งอาจไล่โปรแกรมลงไปลึกมากจนไม่สามารถย้อนกลับไปยังส่วนที่เรียกได้ แต่ในเทอร์โบปาสคาลมีรายการในการค้นหาโพรซีเดอร์ที่ต้องการได้โดยเลือกรายการ DEBUG แล้วเลือกรายการย่อย Call Stack

ทุกครั้งที่โพรซีเดอร์หรือฟังก์ชันถูกเรียกใช้ เทอร์โบปาสคาลจะจดจำส่วนที่เรียกใช้และพารามิเตอร์ที่ส่งมาให้ และเมื่อการทำงานในโพรซีเดอร์หรือฟังก์ชันจบลง จะมีการยกเลิกเนื้อที่ต่าง ๆ สำหรับชื่อและพารามิเตอร์ต่าง ๆ เหล่านี้ ซึ่งขบวนการนี้เรียกว่า Call Stack

เมื่อไรก็ตามที่โปรแกรมหยุดทำงานอาจเกิดจากผู้ใช้ยุติโปรแกรมเอง หรือมาจากการสั่งให้ทำงานทีละคำสั่ง ซึ่งเราอาจขอดูส่วนของสแต็กได้จากรายการ DEBUG และรายการย่อย Call Stack หรือใช้การกดปุ่ม Ctrl กับ F3 แทนได้ จากนั้นจะปรากฏหน้าต่างที่ประกอบด้วยโพรซีเดอร์หรือฟังก์ชันที่กำลังใช้สแต็กอยู่ในขณะนั้น

สมมติว่าเรามีโปรแกรมที่ทำหน้าที่อ่านชื่อเข้ามาแล้วพิมพ์ออกในลักษณะที่เป็น 2 คอลัมน์ โดยที่โปรแกรมที่ทำหน้าที่นี้มีชื่อว่า Address ส่วนโพรซีเดอร์ที่ทำหน้าที่พิมพ์ข้อมูลออกเป็น 2 คอลัมน์นั้นมีชื่อว่า printlist ซึ่งจะเรียกใช้โพรซีเดอร์ที่ชื่อ putleft และ putright เพื่อพิมพ์ข้อมูลของคอลัมน์ทั้งสอง ดังแสดงในรูปที่ 2

ถ้าเรากำหนดจุดจบของโปรแกรมอยู่ในโพรซีเดอร์ที่ชื่อ putleft โดยการเลื่อนเคอร์เซอร์ไปอยู่ในโพรซีเดอร์ดังกล่าว จากนั้นจึงเลือกรายการ RUN และตามด้วยรายการย่อยที่ชื่อว่า Go to cursor เพื่อให้โปรแกรมประมวลผลตั้งแต่ต้นจนกระทั่งถึงตำแหน่งที่เคอร์เซอร์อยู่ จากนั้นจึงเลือกรายการ DEBUG และ Call Stack ทันทีที่โปรแกรมหยุดทำงาน ซึ่งที่จอภาพจะปรากฏดังนี้

```
Call Stack
PUTLEFT ('Allan Jones')
PRINTLIST ('Allan Jones', 100)
ADDRESS
```

จากข้อมูลดังกล่าวบอกให้ทราบว่าโพรซีเดอร์ที่ชื่อ PUTLEFT เป็นโพรซีเดอร์ที่ทำงานอยู่อันดับสุดท้ายก่อนที่โปรแกรมจะหยุดทำงาน ส่วน Allan Jones คือค่าของพารามิเตอร์ที่ถูกส่งมา PUTLEFT ถูกเรียกจากโพรซีเดอร์ที่ชื่อ PRINTLIST ซึ่งโพรซีเดอร์นี้มีการรับชื่อและค่า ๆ หนึ่งที่มีค่า 100 ส่วน PRINTLIST เองถูกเรียกจากโปรแกรมหลักที่ชื่อว่า ADDRESS ถ้าหากโปรแกรมไม่มีชื่อแล้ว ชื่อของโปรแกรมหลักจะเป็น PROGRAM

สิ่งนี้รู้จักกันในนามของสแต็ก เพราะมีการใช้สแต็ก โดยที่แต่ละครั้งที่มีการเรียกใช้โพรซีเดอร์หรือฟังก์ชันแล้ว จะมีการใส่ค่าลงในสแต็ก และเมื่อออกจากโพรซีเดอร์หรือฟังก์ชันจะมีการดึงค่าออกจากสแต็ก

คำสั่ง Call Stack ทำหน้าที่สำคัญอีกอย่างหนึ่งคือ ทำให้สามารถมองย้อนกลับไปตามลำดับที่เรียกใช้โพรซีเดอร์ เมื่อเริ่มเรียกใช้สแต็ก ตำแหน่งบนสุดจะเป็นแถบสว่าง เราสามารถใช้ลูกศรขึ้น-ลงเลื่อนผ่านสแต็กได้ และถ้ากดปุ่ม <ENTER> จะเป็นการเลือกตำแหน่งที่แถบสว่างปรากฏอยู่เป็นครั้งสุดท้ายในโปรแกรมหรือสับรูทีน (Subroutine) นั้น ๆ

```
Program ADDRESS;
type strtype = string[30];
var s : strtype;
    x : integer;

procedure putleft(msg:strtype);
begin
      writeln(msg);
end;

procedure putright(sg:strtype);
begin
      wtiteln('                    ',sg);
end;

procedure printlist(m:strtype;a : integer);
begin
      writeln('Accept value',a);
      putleft(m);
      putright(m);
end;

Begin
      write('Enter name : ');
      readln(s);
      x := 100;'
      printlist (s,x);
End.
```

รูปที่ 2
แสดงโปรแกรมการพิมพ์ในลักษณะที่เป็น 2 คอลัมน์

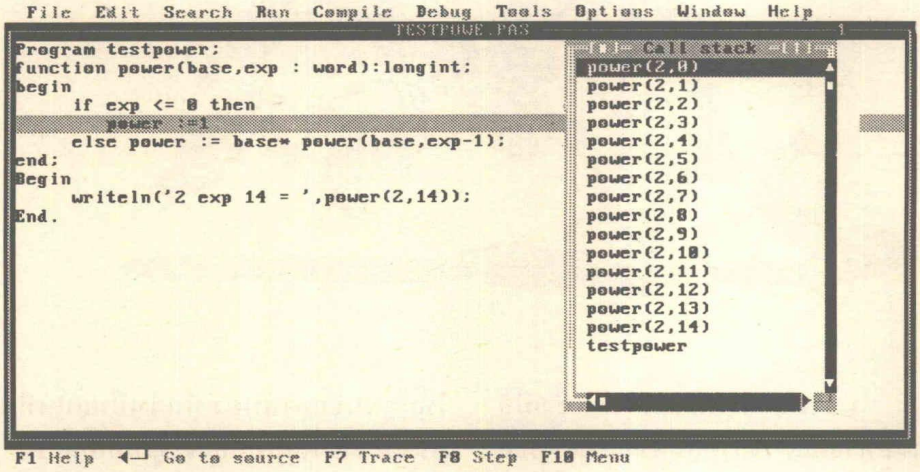
ยกตัวอย่างเช่น จากสแต็กที่ผ่านมาแล้วนั้น ถ้าเลื่อนแถบสว่างไปที่ PRINTLIST แล้วกดปุ่ม <ENTER> เราจะเข้าไปยังตำแหน่งหนึ่งของโพรซีเดอร์ที่ชื่อ PRINTLIST ซึ่งเรียกใช้โพรซีเดอร์ที่ชื่อ PUTLEFT และทำนองเดียวกัน ถ้าเราเลื่อนแถบสว่างไปที่ ADDRESS แล้วกดปุ่ม <ENTER> นั่นหมายความว่าเราอยู่ตรงบริเวณโปรแกรมหลักที่มีการเรียกใช้โพรซีเดอร์ที่ชื่อ PRINTLIST และสุดท้ายก็คือ ถ้าเลือกตำแหน่งบนสุดแล้วกดปุ่ม <ENTER> จะหมายถึงว่าต้องการเข้าไป

```
Program testpower;

function power(base,exp : word):longint;
being
        if exp < = 0 then
            power : = 1
        else power : = base * power(base,exp – 1);
end;

Begin
        writeln('2 exp 14 = ',power(2,14));
End.
```

รูปที่ 3
แสดงตัวอย่างโปรแกรม
ที่จะทำการทดลอง



รูปที่ 4 แสดงหน้าตาของโปรแกรมที่ทำการทดลอง

ยังตำแหน่งที่ถูกประมวลผลสุดท้ายก่อนโปรแกรมจะหยุดทำงาน

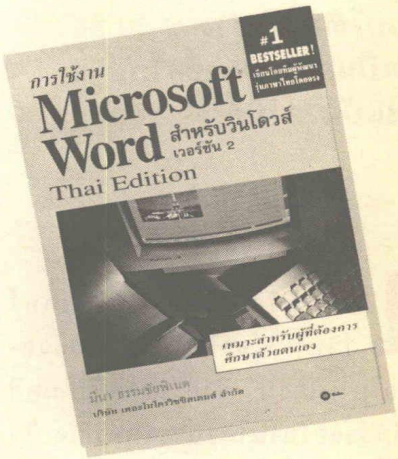
ต่อไปลองพิจารณาโปรแกรมตัวอย่างในรูปที่ 3

ให้พิมพ์โปรแกรมแบบในรูปที่ 3 จากนั้นคอมพายล์จนไม่พบข้อผิดพลาด กำหนดจุดจบของโปรแกรมที่บรรทัดที่สองของโพรซีเดอร์ที่ชื่อ power (ที่บรรทัด power : = 1) แล้วกดปุ่ม F4 เพื่อให้โปรแกรมทำงานจนถึงบรรทัดดังกล่าว เมื่อโปรแกรมหยุดทำงานให้เลือกรายการ DEBUG แล้วตามด้วยรายการย่อย Call Stack จะปรากฏหน้าต่างดังในรูปที่ 4

อย่างไรก็ตามโพรซีเดอร์หรือฟังก์ชันทั้งหมดจะอยู่ในนี้ เราเพียงใช้ลูกศรขึ้น-ลงเพื่อที่จะเลื่อนไปตำแหน่งที่ต้องการเท่านั้น สังเกตว่า Call Stack นี้จะสามารถเก็บชื่อโพรซีเดอร์หรือฟังก์ชันได้สูงสุดเพียง 128 ชื่อเท่านั้น

ตรงนี้

สำหรับผู้ที่
ต้องการศึกษา
ด้วยตนเอง



หนังสือเล่มนี้เขียนขึ้นเพื่อตอบสนองความต้องการของผู้ใช้โปรแกรม Word ที่เพิ่มมากขึ้นตั้งแต่ Word ภาษาไทยออกสู่ท้องตลาด เนื้อหาของหนังสือเริ่มตั้งแต่พื้นฐานการใช้งานวินโดว์จนสามารถใช้งานโปรแกรมได้อย่างสมบูรณ์ เนื่องจากมีเนื้อหาครอบคลุมคำสั่งและเทคนิคการใช้ที่สำคัญ เช่น การจัดรูปแบบตัวอักษรและหน้ากระดาษ การสร้างตาราง การแบ่งคอลัมน์และการนำกราฟิกเข้ามาใช้ในการจัดเอกสาร

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป
หรือที่ ซีเอ็ดบุ๊คเซ็นเตอร์ทุกสาขา
และสั่งซื้อได้ที่ซีเอ็ด สำนักงานใหญ่
โทร 248-6280

มีจำหน่ายแล้ว

ซีเอ็ด